On the EA-Style Integrated Processing of Self-Contained Mathematical Texts

Anatoli I. Degtyarev
Department of Computing and Mathematics,
Manchester Metropolitan University,
e-mail: A.Degtyarev@cs.man.ac.uk

Alexander V. Lyaletski
Cybernetics Department,
Taras Shevchenko University of Kyiv,
e-mail: lav@tk.cyb.univ.kiev.ua

Marina K. Morokhovets
Digital Automata Theory Department,
Glushkov Institute of Cybernetics,
e-mail: mmk@d105.icyb.kiev.ua

Abstract. *In this paper [1], we continue to develop our approach to theorem proof search in the EA-style, that is theorem proving in the framework of integrated processing mathematical texts written in a 1st-order formal language close to the natural language used in mathematical papers. This framework enables constructing a sound and complete goal-oriented sequent-type calculus with "large-block" inference rules. In particular, it contains the formal analogs of such natural proof search techniques as definition handling and auxiliary proposition application. The calculus allows to incorporate symbolic computations in an inference search.*

[1] This research was partially supported by the project INTAS 96-0760 and by the EPSRS grant GR/M46631 given to the first author.

1 Introduction

In early 1960s, V.Glushkov initiated the programme of work on the automation of theorem proof search in mathematics. That programme received later the name "Evidence Algorithm" (EA, or AO due to Russian "Algoritm Ochevidnosti"). One of the basic notions in the EA programme was the notion of evidence of a proof step which changes as EA progresses.

"...The structure of Evidence Algorithm needs to leave room for unbounded replenishment of EA with new and new blocks for the purpose of creating more levels of hierarchy. For practical application of EA, it is important to achieve such a level in its progress when an average length of a proof (including refutation examples construction) comes practically to an average length of proofs required in textbooks and monographs, and then in special papers. Therewith, of course, besides Evidence Algorithm proper, information base of the system which contains descriptions (in practical mathematical logic language) of different kinds of concepts used in a specific mathematical theory under consideration, as, also, properties of these concepts, procedures for the generation and investigation of examples, etc is to be developed. All this information abundance ought to be used by EA just as a human does it." [1]

In accordance with the principles [2] of this programme, the following scheme of automated theorem proving in the EA-style have been developed. An assertion to be proven is immersed in a mathematical text written in so-called Theory Language (TL) [3]. The language TL was designed to meet the following requirements: to be a communication link between a user and a computer within an automated theorem proving system; to be a formal language for representing and storing mathematical data; to be a high-level language close to a language of usual mathematical publications. The language TL contains a rich collection of linguistic structures that, on the one hand, have precise definitions (in terms of BNF), and, on the other hand, are similar to constructions of natural languages. That is why it has been selected as the first approximation to a family of formalized mathematical languages for EA.

Then, such a TL-text is transformed into a so-called $TL1$-text, if it is possible. A $TL1$-text consists of $TL1$-sentences, which are, on the one hand, analogs of 1st-order logic formulas, and, on the other hand, preserve the signature of an original TL-text, its syntax and structure (i.e. partitioning into definition sections, auxiliary proposition sections and a theorem to be proved). In what follows, the dual nature of a $TL1$-sentence will be often exploited. Now a $TL1$-text is a source for proof search in the EA-style.

In the framework of such an approach to automated theorem proving in the EA-style, a special sequent-type formalism has been developed [4]. As a result, two so-called a-sequent calculi have been suggested [5]. They are sound and complete and satisfy the following requirements: syntactical form of an initial problem should be preserved; deduction should be done in the signature of initial texts, in particular, preliminary skolemization should be non-obligatory; proof searching should be goal-oriented; equality handling should be separated from deductive processes.

In this paper, we continue to develop our approach to theorem proof search in the EA-style, that is theorem proving in the framework of integrated processing mathematical texts written in a formal 1st-order language close to the natural language used in mathematical papers. This framework enables constructing a sound and complete goal-oriented sequent-type calculus with “large-block” inference rules. In particular, it contains the formal analogs of such natural proof search techniques as definition handling and auxiliary proposition application.

1.1 Basic computer-oriented approaches to logical inference search in 1st-order classical calculi

The beginning of EA programme realization can be referred to the publication of the paper [6] on the heuristic procedure for theorem proof search in Group Theory in which there was made allowance (on a formal level) for some proof search methods used in mathematical papers. Then that formal technique was extended to specific fragments of Set Theory. Its final completion appeared as the specific calculus AGS (Auxiliary Goal Search) [7] which was meant for ascertaining the validity of 1st-order classical logic formulas. In parallel, research on a formal language [8] oriented to the representation of mathematical texts intended to be processed by a computer was performed. That language can be viewed as an analog of a classical 1st-order language enriched with means which enable to use constructions and notation more convenient and usual for mathematicians.

Historically, it is worth noting that the end of the 1950s - beginning of the 1960s can be characterized as the period of coming into existence computers with such a performance rate, information capacity, and flexibility that programming complex intellectual processes became feasible. As a response to the emergence of computing machinery of that sort a series of papers appeared in which the issues of the implementation of Gentzen-type calculi [9] and inference search methods relying on the results of Skolem [10] and Herbrand [11] were discussed. For more detail, the reader is referred to, for example, [12], [13], [14], [15], etc. It might be well to point out that in those first papers an answer to the principal question about a possibility to use computers for mathematical reasoning was provided. But the lack of machine-oriented techniques for the optimization of enumeration during reasoning impeded getting a proof even for rather simple true assertions in 1st-order classical logic. Investigations with the aim of improving the efficiency of the proof search methods proposed resulted in coming into existence Kanger's calculus [16] (of Gentzen type) and Robinson's resolution method [17] (of Skolem-Herbrand type). For those days, the latter yielded the most efficient machine-oriented inference search technique for the 1st-order classical predicate calculus by using Robinson's unification algorithm. (S.Yu.Maslov's inverse method [18] is worth of special noting here. It can be formulated in resolution terms for 1st-order classical logic, although the initially proposed scheme of the inverse method was subsequently extended to non-classical logics.) Thus studies on automated theorem proving were later on concentrated mainly upon improving the potentialities of the Skolem-Herbrand approach. So, unification algorithms which took account of specific features of a particular 1st-order theory with equality were proposed (A-unification, AC-unification, etc.). The ways of building-in special equality handling rules (for instance, the paramodulation rule) into resolution-based methods have been investigated. Subsequent attempts for the advancement of the Skolem-Herbrand approach resulted in arising tableaux methods, the connection graph method, goal-oriented search methods, etc.

The lower efficiency of Kanger's approach (as compared with the Skolem-Herbrand approach) can be explained by the fact that Kanger-type calculi (and the AGS calculus is among them) do not need the obligatory carrying out of preliminary skolemization that can result in arising the superfluous enumeration caused by the possibility of different orders of logical (mainly, quantifier) inference rules application and the necessity of center formulas duplication when applying some of those rules. At the same time, machine-oriented Gentzen-Kanger-type methods reflect proof techniques which are more "natural" for a human. They also enable to construct rather flexible tools to support a dialogue between a user and a computer during interactive inference search, and to facilitate understanding a proof process by a user. So, in the framework of EA programme realization activity, achieving the improved search efficiency of AGS by a possibility both of "transition" to the sound (and complete, if possible) deductive processing of mathematical texts written in a formalized mathematical language close to usual mathematical publication languages and the application of mathematical facts gained came forward as a central problem of proof search automation.

As a result of research performed, the sound and complete 1st-order calculus of a-sequents with an original notion of an admissible substitution has been suggested [4]. This notion has been introduced with the aim of the optimization of additional efforts connected with the possibilities of different orders of quantifier elimination without preliminary skolemization. (It has been shown later that the notion of an admissible substitution can be easily "built-in" into standard Gentzen calculi [19].)

Along with Gentzen-Kanger-type calculi, Skolem-Herbrand-type calculi have been investigated [20]. The retrospective point of view on the linguistic tools and deductive systems of EA can be found in [21].

1.2 The current state of the EA-style inference search

Nowadays we carry out research work within the EA programme (see Website: http://tc.cyb.univ.kiev.ua/ea/) at a new level of understanding the problem of automated theorem proving and taking into account existing trends in the development of program systems which support "doing mathematics".

In [5], the a-sequent calculi (gS and mS) were proposed, and the formal description of gS was given, but the calculus mS was only illustrated by an example, whereas the inference rules of mS were not presented. To make up this deficiency, we give in this paper the formal description of mS. This paper reflects continuing efforts to attack the problem of automated theorem-proving in the EA-style by the application of definitions and auxiliary propositions. Following [5], we denote the corresponding calculus as mS. To make the paper "self-contained" enough we give here the necessary definitions.

The calculus mS permits to present an initial problem as a text in a certain 1st-order formal language containing definitions and auxiliary propositions, and to use analogs of such natural theorem proving techniques as the application of definitions and auxiliary propositions. The peculiarity of our approach is that needed definitions and auxiliary propositions are extracted from a self-contained mathematical text written in the formal language TL [3] approximated to languages of usual mathematical papers. A self-contained mathematical text is a text that, in addition to a proposition to be proved, also includes assumptions, propositions, and definitions that can be used when the proof of a given assertion is searching for.

Processing a self-contained mathematical text for the purpose of proving a given theorem is divided into three parts:

(1) writing down an original mathematical text as a TL-text;

(2) translating the TL-text into a $TL1$-text;

(3) searching for a proof in mS within the $TL1$-text environment.

2 The calculus mS and theorem proving

In this section, we present the calculus mS as a deductive basis for solving the problem of the validity of a given assertion in the context of a natural mathematical text. After the text is written in TL-language and converted into a $TL1$-text, a theorem proof search is carried out using the inference rules of mS.

The calculus mS is an extension of gS [5] with additional inference rules for the application of definitions and auxiliary propositions.

2.1 Preliminaries

The basic object of mS is an a-sequent. An a-sequent may be considered as a special generalization of the standard notion of a sequent. We consider a-sequents having one object ("goal") in its succedent only.

We treat here 1st-order classical logic in the form of the sequent calculus G given in [22].

We treat the notion of a substitution as in [17]. Any substitution component is considered to be of the form t/x, where x is a variable and t is a term of a substitution.

Let L be a literal, then $\sim L$ denotes its complement. We use the expression $L(t_1, \ldots, t_n)$ to denote that t_1,...,t_n is a list of all the terms (possibly, with repetitions) occupying the argument places in the literal L in the order of their occurrences in L. If x, y are variables and F is a formula then $F|_y^x$ denotes the result of replacing x with y.

We also assume that besides usual variables there are two countable sets of special variables, namely unknown variables and fixed variables (dummies and parameters in the terminology of [16]).

An ordered triple $< w, F, E >$ is called *an ensemble* iff w is a sequence (a word) of unknown and fixed variables, F is a 1st-order formula, and E is a set of pairs of terms t_1, t_2 (equations of the form $t_1 = t_2$).

An a-sequent is an expression of the form $[\mathcal{P}], [\mathcal{D}], [B], < w_1, P_1, E_1 >$,..., $< w_n, P_n, E_n > \Rightarrow < w, F, E >$, where $< w_1, P_1, E_> $,...,$< w_n, P_n, E_n >$, $< w, F, E >$ are ensembles, $[B]$ is a list of literals, possibly empty, $[\mathcal{P}]$ and $[\mathcal{D}]$ are lists of $TL1$-sentences corresponding to auxiliary propositions and definitions, respectively.

Ensembles in the antecedent of an a-sequent are called premises, and an ensemble in the succedent of an a-sequent is called a goal of this a-sequent. The collection of the premises is thought as a set. So, the order of the premises is immaterial.

Let W be a set of sequences of unknown and fixed variables, and s be a substitution. Put $A(W,s) = \{< z,t,w >$: z is a variable of s, t is a term of s, $w \in W$, and z lies in w to the left of some fixed variable from $t\}$. Then s is said to be *admissible* for W iff (1) the variables of s are unknown variables only, and (2) there are not (different) elements $< z_1, t_1, w_1 >$,...,$< z_n, t_n, w_n >$ in $A(W,s)$ such that $t_2/z_1 \in s$,...,$t_n/z_{n-1} \in s$,$t_1/z_n \in s$ $(n > 0)$.

Decomposition of some $TL1$-sentence F by its principal logical connective and possible interaction with P_i results in generating new a-sequents. The sets E_1,...,E_n, E define the terms to be substituted for the unknown variables in order to transform every equation $t_1 = t_2$ from E_1,...,E_n, E to identity $t = t$ after applying to E_1,...,E_n, E a substitution chosen in a certain way. The sets w_1,...,w_n, w serve to check whether the substitutions generated during proof searching are admissible. Note, that in any a-sequent some (or all) sequences from w_1,...,w_n, w and some (or all) sets from E_1,...,E_n, E may be empty.

An initial a-sequent is constructed as follows. Suppose, a self-contained $TL1$-text Txt consists of the collection $[\mathcal{D}]$ of definitions and the collection $[\mathcal{P}]$ of auxiliary propositions, and a theorem T is given, which can be interpreted in terms of the calculus G as a sequent of the form $P_1, \ldots, P_n \Rightarrow F$. Then an a-sequent $[\mathcal{P}], [\mathcal{D}], [], <, P_1, >, \ldots, <, P_n, > \Rightarrow <, F, >$ will be considered as an initial a-sequent (w.r.t. T and Txt).

During proof searching in mS an inference tree is constructed. At the beginning of a search process it consists of an initial a-sequent. The subsequent nodes of the inference tree are generated in accordance with the rules described below. Inference trees grow "from top to bottom".

2.2 The calculus mS

In the formulation of rules below, M denotes a set of premises, $[\mathcal{D}]$ ($[\mathcal{D}_1]$, $[\mathcal{D}_2]$) is a list of definitions, possibly empty, $[\mathcal{P}]$ ($[\mathcal{P}_1]$, $[\mathcal{P}_2]$) is a list of auxiliary propositions, possibly empty. Sometimes, $[\mathcal{P}]$, $[\mathcal{D}]$ will be omitted when they are not used.

Let us introduce inductively a notion of *a positive (negative) occurrence of a literal L in a formula F (denoted by $F\lfloor L^+ \rfloor$ and $F\lfloor L^- \rfloor$, respectively) modulo equations* in a rigorous way:
(I) suppose that a literal F ($\sim F$) can be obtained from $L(t_1, \ldots, t_n)$ by means of replacing t_1,...,t_n with some terms t'_1,...,t'_n. Then L is said to have a positive (negative) occurrence in the literal F modulo the equations $t_1 = t'_1$,...,$t_n = t'_n$;
(II.1) if $F\lfloor L^+ \rfloor$ ($F\lfloor L^- \rfloor$) modulo the equations $t_1 = t'_1, \ldots, t_n = t'_n$ and

F_1 is a formula then L has a positive (negative) occurrence (modulo the equations $t_1 = t'_1, \ldots, t_n = t'_n$) in the following formulas: $F \wedge F_1$, $F_1 \wedge F$, $F \vee F_1$, $F_1 \vee F$, $F_1 \supset F$, $\forall x F$, $\exists x F$;
(II.2) if $F\lfloor L^+ \rfloor$ ($F\lfloor L^- \rfloor$) modulo the equations $t_1 = t'_1, \ldots, t_n = t'_n$ and F_1 is a formula then L has a negative (positive) occurrence (modulo the equations $t_1 = t'_1, \ldots, t_n = t'_n$) in the following formulas: $F \supset F_1$, $\neg F$;
(III) there are no other cases of positive (negative) occurrences of L in F.

Goal splitting rules (GS)

These rules are used for the elimination of the principal logical connective from the $TL1$-sentence in the goal of an a-sequent processed. The application of any rule results in generation of a new a-sequent with only one goal (and, possibly, with new premises). The elimination of the $TL1$-equivalents of proposition connectives is done according to 1st-order classical logic (it can be easily expressed in the terms of derivative rules of standard Gentzen-type calculi [22]), and $w_1,\ldots,w_n, w, E_1,\ldots,E_n$ therewith are not changed. Essential deviation from traditional Gentzen inference search techniques is observed in the processing of quantifiers. This deviation reflects specific quantifier handling techniques investigated in [4], where variables of eliminated quantifiers are replaced by undefined or fixed variables depending on an eliminated quantifier. Therewith w, but not $w_1,\ldots,w_n, E_1,\ldots,E_n, E$, is changed, and new premises can be generated.
Propositional Rules
($\Rightarrow\supset_1$)-rule:

$$\frac{[B], M \Rightarrow < w, F \supset F_1, E >}{[B], M, < w, F, E > \Rightarrow < w, F_1, E >}$$

($\Rightarrow\supset_2$)-rule:

$$\frac{[B], M \Rightarrow < w, F \supset F_1, E >}{[B], M, < w, \neg F_1, E > \Rightarrow < w, \neg F, E >}$$

($\Rightarrow \wedge$)-rule:

$$\frac{[B], M \Rightarrow < w, F \wedge F_1, E >}{[B], M \Rightarrow < w, F, E > \quad [B], M \Rightarrow < w, F_1, E >}$$

($\Rightarrow \vee_1$)-rule:

$$\frac{[B], M \Rightarrow < w, F \vee F_1, E >}{[B], M, < w, \neg F, E > \Rightarrow < w, F_1, E >}$$

($\Rightarrow \vee_2$)-rule:

$$\frac{[B], M \Rightarrow < w, F \vee F_1, E >}{[B], M, < w, \neg F_1, E > \Rightarrow < w, F, E >}$$

($\Rightarrow \neg$)-rule:

$$\frac{[B], M \Rightarrow < w, \neg F, E >}{[B], M \Rightarrow < w, F', E >}$$

where F' is the result of one-step transferring "$\neg$" into F.

Quantifier Rules

($\Rightarrow \forall$)-rule:

$$\frac{[B], M \Rightarrow < w, \forall x F, E >}{[B], M \Rightarrow < w\overline{x}, F|^{x}_{\overline{x}}, E >}$$

where $\overline{x}$ is a new fixed variable.

($\Rightarrow \exists$)-rule:

$$\frac{[B], M \Rightarrow < w, \exists x F, E >}{[B], M, < w, \forall x \neg F, E > \Rightarrow < wx', F|^{x}_{x'}, E >}$$

where x' is a new unknown variable.

Auxiliary goal rules (AG)

These rules reflect the fact that mS is oriented to proof searching by certain transformation of the sentences of self-contained mathematical $TL1$-texts. In terms of Gentzen-type calculi, AG-rules can be interpreted as the elimination of principal logical connectives from $TL1$-sentences, which occur in premises. (Those sentences are generated deterministically beginning with such a premise $< w_i, P_i, E_i >$ that P_i contains a positive occurrence (modulo equations) of the $TL1$-sentence F from the goal of an input a-sequent for AG. As to the elimination of principal logical connectives in premises, the remarks referring to GS are true, excluding, naturally, remarks on w_1,...,w_n, w, E_1,...,E_n, E.) The application of AG results in generation of m $(m > 0)$ a-sequents with new goals $< w'_1, F_1, E'_1 >$,...,$< w'_m, F_m, E'_m >$ and, possibly, some new (w.r.t. the input a-sequent for AG) premises.

Propositional Rules

($\supset_1 \Rightarrow$)-rule:

$$\frac{[B], < w, F\lfloor L^- \rfloor \supset F_1, E' >, M \Rightarrow < w', L, E >}{[B], < w, (\neg F)\lfloor L^+ \rfloor, E' >, M \Rightarrow < w', L, E > \quad [B, \sim L], M \Rightarrow < w, \neg F_1, E' >}$$

($\supset_2 \Rightarrow$)-rule:

$$\frac{[B], < w, F \supset F_1\lfloor L^+ \rfloor, E' >, M \Rightarrow < w', L, E >}{[B], < w, F_1\lfloor L^+ \rfloor, E' >, M \Rightarrow < w', L, E > \quad [B, \sim L], M \Rightarrow < w, F, E' >}$$

($\vee_1 \Rightarrow$)-rule:

$$\frac{[B], < w, F \vee F_1\lfloor L^+ \rfloor, E' >, M \Rightarrow < w', L, E >}{[B], < w, F_1\lfloor L^+ \rfloor, E' >, M \Rightarrow < w', L, E > \quad [B, \sim L], M \Rightarrow < w, \neg F, E' >}$$

($\vee_2 \Rightarrow$)-rule:

$$\frac{[B], < w, F\lfloor L^+ \rfloor \vee F_1, E' >, M \Rightarrow < w', L, E >}{[B], < w, F\lfloor L^+ \rfloor, E' >, M \Rightarrow < w', L, E > \quad [B, \sim L], M \Rightarrow < w, \neg F_1, E' >}$$

($\wedge_1 \Rightarrow$)-rule:

$$\frac{[B], < w, F\lfloor L^+ \rfloor \wedge F_1, E' >, M \Rightarrow < w', L, E >}{[B], < w, F\lfloor L^+ \rfloor, E' >, < w, F_1, E' >, M \Rightarrow < w', L, E >}$$

($\wedge_2 \Rightarrow$)-rule:

$$\frac{[B], < w, F \wedge F_1\lfloor L^+ \rfloor, E' >, M \Rightarrow < w', L, E >}{[B], < w, F, E' >, < w, F_1\lfloor L^+ \rfloor, E' >, M \Rightarrow < w', L, E >}$$

($\neg \Rightarrow$)-rule:

$$\frac{[B], < w, \neg(F\lfloor L^- \rfloor), E' >, M \Rightarrow < w', L, E >}{[B], < w, F'\lfloor L^+ \rfloor, E' >, M \Rightarrow < w', L, E >}$$

where F' is the result of one-step transferring "$\neg$" into F.
Termination Rules
($\Rightarrow \sharp_1$)-rule:

$$\frac{[B], < w, L(t_1, \ldots, t_n), E' >, M \Rightarrow < w', L(t'_1, \ldots, t'_n), E >}{M \Rightarrow < w, \sharp, E'' >}$$

(Here $E'' = E' \cup E \cup \{t_1 = t'_1, \ldots, t_n = t'_n\}$; $L(t_1, \ldots, t_n)$, $L(t'_1, \ldots, t'_n)$ are literals.)
($\Rightarrow \sharp_2$)-rule:

$$\frac{[B_1, L(t_1, \ldots, t_n), B_2], M \Rightarrow < w', L(t'_1, \ldots, t'_n), E >}{[B_1, L(t_1, \ldots, t_n), B_2], M \Rightarrow < w', \sharp, E' >}$$

(Here $E' = E \cup \{t_1 = t'_1, \ldots, t_n = t'_n\}$; $L(t_1, \ldots, t_n)$, $L(t'_1, \ldots, t'_n)$ are literals.)
Quantifier Rules
($\forall \Rightarrow$)-rule:

$$\frac{[B], < w, \forall x(F\lfloor L^+ \rfloor), E' >, M \Rightarrow < w', L, E >}{[B], < wx', F|^x_{x'}\lfloor L^+ \rfloor, E' >, < w, \forall x F, E' >, M \Rightarrow < w', L, E >}$$

where x' is a new unknown variable.
$(\exists \Rightarrow)$-rule:

$$\frac{[B], < w, \exists x(F\lfloor L^+\rfloor), E' >, M \Rightarrow < w', L, E >}{[B], < w\overline{x}, F|_{\overline{x}}^{x}\lfloor L^+\rfloor, E' >, M \Rightarrow < w', L, E >}$$

where $\overline{x}$ is a new fixed variable.

Definition application rule and auxiliary proposition rule

Structuring $TL1$-texts according to substantive sections (i.e. definitions, propositions, etc.) enables introducing in mS the *definition application rule* (DA) and the *auxiliary proposition rule* (AP) in a natural way. These rules can be viewed as specific variants of AG. They represent analogs of natural theorem-proving techniques for the application of definitions and auxiliary propositions.

The rule DA is formulated in view of the structure of the definition section of a TL-text. According to the syntax of the language TL, a definition section consists of two parts, namely, a description part (or description) and a definition part. A description part is a collection of assumptions which satisfies the following closure condition: for any variable x that has an occurrence in some assumption from the description there exists an assumption (in the same description) of the form "Let x be K" or "$x \in M$ ", where K is a concept, M is a term. A definition part is a TL-sentence of the form $A(x_1, \ldots, x_n)$ IFF $\mathcal{F}(x_1, \ldots, x_n)$, where x_1,...,x_n are variables and every x_i $(i = 1, \ldots, n)$ has an occurrence in the description part. A formula $A(x_1, \ldots, x_n)$, as a rule, is an atom, and $\mathcal{F}(x_1, \ldots, x_n)$ is a formula that does not include the atom A (if we restrict ourselves to non-recursive definitions).

A definition section with a description part consisting of the assumptions F_1,...,F_k and with a definition part of the form $A(x_1, \ldots, x_n)$ IFF $\mathcal{F}(x_1, \ldots, x_n)$ can be viewed as the formula $\forall x_1 \ldots x_n(F_1 \wedge \ldots \wedge F_k \supset (A(x_1, \ldots, x_n) \equiv \mathcal{F}(x_1, \ldots, x_n)))$. Since the application of a definition in a mathematical practice is generally done as if one of the formulas, $\forall x_1 \ldots x_n(F_1 \wedge \ldots \wedge F_k \supset (A(x_1, \ldots, x_n) \supset \mathcal{F}(x_1, \ldots, x_n)))$ or $\forall x_1 \ldots x_n(F_1 \wedge \ldots \wedge F_k \supset (\mathcal{F}(x_1, \ldots, x_n) \supset A(x_1, \ldots, x_n)))$, were used, we treat a definition section as a pair of the above formulas.

An auxiliary proposition is a TL-text section consisting of a collection of assumptions and a conclusion. The conclusion can be viewed as a formula $F(x_1, \ldots, x_n)$, where x_1,...,x_n are variables occurring in the assumptions of the proposition. The collection of assumptions satisfies the above closure condition or it can be transformed in such a way to satisfy the closure condition (we do not consider this transformation here, see, for example [23], [20] on this issue).

In DA-rule below, $F\lfloor L^+\rfloor$ denotes either $(A(x_1,\ldots,x_n)\supset\mathcal{F}(x_1,\ldots,x_n))$ or $(\mathcal{F}(x_1,\ldots,x_n)\supset A(x_1,\ldots,x_n))$.

DA-rule:

$$\frac{[\mathcal{P}],[\mathcal{D}_1,\forall x_1\ldots x_n(F_1\wedge\ldots\wedge F_k\supset F\lfloor L^+\rfloor),\mathcal{D}_2],[B],M\Rightarrow<w',L,E>}{\mathcal{A}_1\Rightarrow<w',L,E>\qquad\mathcal{A}_1'\Rightarrow<x_1'\ldots x_n',F_1\wedge\ldots\wedge F_k,>}$$

(Here $\mathcal{A}_1$ denotes $[\mathcal{P}],[\mathcal{D}_1,\forall x_1\ldots x_n(F_1\wedge\ldots\wedge F_k\supset F\lfloor L^+\rfloor),\ \mathcal{D}_2],\ [B],$ $<x_1'\ldots x_n',\ F\lfloor L^+\rfloor,>,M$; $\mathcal{A}_1'$ denotes $[\mathcal{P}],\ [\mathcal{D}_1,\forall x_1\ldots x_n(F_1\wedge\ldots\wedge F_k\supset F\lfloor L^+\rfloor),\ \mathcal{D}_2],\ [B,\sim L],M$.)

AP-rule:

$$\frac{[\mathcal{P}_1,\forall x_1\ldots x_n(F_1\wedge\ldots\wedge F_k\supset F\lfloor L^+\rfloor),\mathcal{P}_2],[\mathcal{D}],[B],M\Rightarrow<w',L,E>}{\mathcal{A}_2\Rightarrow<w',L,E>\qquad\mathcal{A}_2'\Rightarrow<x_1'\ldots x_n',F_1\wedge\ldots\wedge F_k,>}$$

(Here $\mathcal{A}_2$ denotes $[\mathcal{P}_1,\forall x_1\ldots x_n(F_1\wedge\ldots\wedge F_k\supset F\lfloor L^+\rfloor),\ \mathcal{P}_2],[\mathcal{D}],[B],<$ $x_1'\ldots x_n',F\lfloor L^+\rfloor),>,M$; $\mathcal{A}_2'$ denotes $[\mathcal{P}_1,\forall x_1\ldots x_n(F_1\wedge\ldots\wedge F_k\supset F\lfloor L^+\rfloor,$ $\mathcal{P}_2],\ [\mathcal{D}],[B,\sim L],M$.)

Premise addition rule (PA-rule)

This rule affects the whole proof search tree. After every application of ($\forall\ \Rightarrow$)-rule (($\exists\ \Rightarrow$)-rule), the new premise $<wx',\ F|^x_{x'}\lfloor L^+\rfloor,\ E'>$ ($<w\overline{x},F|^x_{\overline{x}}\lfloor L^+\rfloor,E'>$) is added to antecedents of all a-sequents containing a premise with a formula which includes the marked occurrence of $<w,$ $\forall x(F\lfloor L^+\rfloor),\ E'>$ ($<w,\exists x(F\lfloor L^+\rfloor),E'>$) through the current tree.

Axioms

Axioms are a-sequents of the form $[\mathcal{P}],[\mathcal{D}],[B],M\Rightarrow<w,\sharp,E>$, where $\sharp$ denotes an empty formula.

An inference tree

The assertion T to be proved is represented as a substantive $TL1$-section "theorem", in which conditions, or assumptions, and a conclusion are separated, and an initial a-sequent (with respect to T) is constructed with the assumptions and conclusion in its antecedent and succedent, respectively. (The remained part Txt of the $TL1$-text is given as the set of definitions and auxiliary propositions.)

Applying the rules "from top to bottom" to the input a-sequent and afterwards to its "heirs", and so on, we finally obtain an *inference tree* (w.r.t. the theorem T to be proved and "environmental" $TL1$-text Txt). The inference tree Tr is called a *proof tree* for an input a-sequent if and only if (1) every leaf of Tr is an axiom; (2) there exists the unifier s of all equations from Tr; (3) s is admissible (in the sense of this paper) for the set of all sequences of fixed and unknown variables from the leaves of Tr.

At any moment during inference search, it is possible to test whether a current inference tree can be transformed into a proof tree. When the construction of a proof tree is made in an interactive mode, a user may initiate this test. For testing, techniques for finding the most general unifier can be used.

3 Main results

It was noted above that any $TL1$-sentence can be treated as an analog of some 1st-order classical logic formula. It enables constructing formula patterns of such units of a $TL1$-text as the theorem to be proved, a definition, an auxiliary proposition and to treat a self-contained $TL1$-text as a set of 1st-order formulas. So, it is possible to understand unambiguously such terms as "$TL1$-text consistency", "logical consequence of a theorem from a given $TL1$-text", and "validity" (of the theorem to be proved) without special defining the semantics of the $TL1$-language. With this in mind, we state main results about mS as follows.

Proposition 1 (soundness and completeness of mS). $TL1$-theorem T is a logical consequence of a consistent $TL1$-text Txt (which does not include T) if and only if a proof tree (with an initial sequent w.r.t. T and Txt) can be constructed in mS.

Proposition 2. A $TL1$-theorem T is valid if and only if a proof tree (with an initial sequent w.r.t. T only) can be constructed in mS.

We note, as a side-result, that rather rich collection of rules in mS enables to construct various proof search strategies which model proofs from usual mathematical texts, and, by maintaining the interactive mode of proof search, to allow a user to influence a proof process actively. If such a strategy (with or without participation of a human) ensures an exhaustive search, then propositions 1 and 2 guarantee the soundness and completeness of the strategy.

4 Conclusion

The described approach to logical inference search enables to gain the following benefits.

1. Through using an original notion of an a-sequent instead of a standard notion of a sequent it is possible to develop a quantifier handling technique (without preliminary skolemization) that allows achieving inference search efficiency comparable with the efficiency of methods requiring preliminary skolemization.

2. A-sequents allow to reduce, by rather standard logical transformation, the assertion to be proven to a number of new auxiliary assertions without specifying which terms should be substituted instead of variables.

3. Forming the collections of equations (which can be treated as certain constraints) during proof search allows to postpone finding a solution for equations up to arbitrary moment of time, to separate it from deductive process, and then to use various equation solving techniques (for example, standard unification, AC-unification, E-unification), to built-in specific equality handling rules (e.g., paramodulation), and to apply rewriting techniques and various tools of computer algebra systems.

In view of the item 3, it is possible to say that we have all the necessary to begin the integration of deductive procedures with computer algebra systems.

As proof search methods relying on the principles described above can operate within the signature of an input problem, so it is possible to generate sufficient assertions and derive consequences in the form, which is usual for man. This enables to construct flexible interface tools.

Nowadays, there exists a series of projects and systems which have intersections with the EA programme in one way or another and which are close to it in spirit, for example, MIZAR, THEOREMA, OMEGA, ISABELLE, QED. A number of the above and some other projects, systems, and groups (for example, the DReaM group, Mechanized Reasoning Group, CAAR group, etc.) are interested in the integration of the deductive and computational power of both deduction systems and computer algebra systems. Taking this fact into consideration, the authors hope that this paper and some theses on the EA programme can be helpful in attacking such problems as distributed automated theorem proving, checking self-contained mathematical texts for correctness, remote training in mathematical disciplines, extracting knowledge from mathematical papers, and constructing data bases for mathematical theories.

References

[1] Glushkov, V. Foundations of mathematics and automation of deductive constructing (in Russian). In: Kibernetika. Voprosy teoriyi i praktiki. Nauka, Moskva, 1986.

[2] Lyaletski, A., Morokhovets M. On linguistic aspects of integration of computer mathematical knowledge. In: Proc. of CALCULEMUS Workshop, Trento, Italy (1999) 165–178.

[3] Glushkov, V., Vershinin, K., et al. On a formal language for description of mathematical texts (in Russian). In: Avtomatizatsiya poiska dokazatel'stv teorem v matematike. Institute of Cybernetics, Kiev (1974) 3–36.

[4] Degtyarev, A., Lyaletski, A. Logical inference in SAD (in Russian). In: Matematicheskiye osnovy sistem iskusstvennogo intellekta, Institute of Cybernetics, Kiev, 1981.

[5] Degtyarev, A., Lyaletski, A., Morokhovets, M. Evidence Algorithm and Sequent Logical Inference Search. In: LNAI **1705** (1999) 44–61.

[6] Anufriyev, F., Fediurko, V., Letichevski, A., Asel'derov, Z., Didukh, I. On one algorithm of theorem proof search in Group Theory (in Russian). Kibernetika, N1, 1966, pp. 23 - 29.

[7] Anufriyev, F. An algorithm of theorem proof search in logical calculi (in Russian). In: Teoriya avtomatov, N 5, Institute of Cybernetics, Kiev, 1969.

[8] Glushkov, V., Kostyrko, V., Letichevski, A., Anufriyev, F., Asel'derov, Z. On a language for description of formal theories (in Russian). In: Teoreticheskaya kibernetika, N 3, 1970.

[9] Gentzen, G. Untersuchungen uber das Logische Schliessen. Math. Z. N 39 (1934) 176–210.

[10] Skolem,T. Logisch-kombinatorische Untersuchungen uber die Erfullbarkeit oder Beweisbarkeit mathematischer Satze. Skriftner utgit ar Videnskapsselskaper i Kristiania, **4** (1920) 4–36.

[11] Herbrand, J. Recherches sur la Theorie de la Demonstration. Travaux de la Societe des Sciences et de Lettres de Varsovie, Class III, Sciences Mathematiques et Physiques (1930) Vol. 33.

[12] Davis, M., Putnam, H. A computing procedure for quantification theory. In: J. of the ACM **7** (1960) 201–215.

[13] Gilmore, P.S. A proof method for quantification theory; its justification and realization. In: I.B.M.J. Res. Devl. **4** (1960) 28–35.

[14] Prawitz, D. An improved proof procedure. In: Theoria **26** (1960) 102–139.

[15] Wang Hao Towards mechanical mathematics. In: I.B.M.J. Res. Devl. **4** (1960) 2–22.

[16] Kanger, S. Simplified proof method for elementary logic. In: Comp. Program. and Form. Sys.: Stud. in Logic. North-Holl., Publ. Co., 1963.

[17] Robinson, J. A machine-oriented logic based on resolution principle. In: J. of the ACM (1965) 23–41.

[18] Maslov, S. An inverse method for deducibility checking in classical predicate calculus. In: Doklady Akademiyi Nauk SSSR **159** (1964).

[19] Lyaletski, A. Gentzen calculi and admissible substitutions. In: Actes preliminaries, du Simposium Franco-Sovetique "Informatika-91". Grenoble, France (1991) 99–111.

[20] Atayan, V., Morokhovets, M. Combining formal derivation search procedures and natural theorem proving techniques in an automated theorem proving system. Cybernetics and System Analysts **32** (1996) 442–465.

[21] Degtyarev, A., Kapitonova, Yu., Letichevsky, A., Lyaletski, A., Morokhovets, M. A brief historical sketch on Kiev school of automated theorem proving. In: Proc. of the 2nd International THEOREMA Workshop, Linz, Austria (June 1998) 151–156.

[22] Gallier, J. Logic for computer science: foundations of Automatic Theorem Proving. Harper and Row, Inc. New York (1986) 513 p.

[23] Atayan, V., Vershinin, K. On formalization of some inference search methods (in Russian). In: Avtomatizatsiya obrabotki matematicheskikh tekstov. Institute of Cybernetics, Kiev (1980) 36–52.

[24] Davis, M. Applied non-standard analysis. (Translated from English). Moskva, Mir (1986) 264 p.

5 Appendix: Example

In what follows, we exemplify an inference in mS by presenting a proof (but not a proof search) of a theorem chosen from [24]: "Any closed subset of a compact set is compact". According to the EA-style proof search scheme, that theorem should be "immersed" in a mathematical text. We present here the part of the text that includes assumptions and sections really used in the course of formal inference search. So, some notions (for example, "infinitesimally close", the function ST) have no definitions in the text below, as those definitions are not used as the premises of DA-rule. We use a standard Gentzen notation for sequents instead of a-sequent notation, "hiding" equation handling and variable sequence processing into comments.

The theorem to be proven is treated as a part of the following self-contained mathematical TL-text.

1. "Sets and Relations"

Lemma 1. Let V be a set. Let W be a subset of V. W is a set.

Lemma 2. Let V be a set. Let U be a subset of V. If t is an element of U then t is an element of V.

Lemma 3. Let Y be a set. $ST(Y)$ is a set.

Lemma 4. Let U be a subset of X. Let V be a subset of X. If U is a subset of V then $ST(U)$ is a subset of $ST(V)$.

Lemma 5. "The transitivity property of inclusion."

For any U, V, W it_is_true_that if U is a subset of V and V is a subset of W then U is a subset of W.

2. "Topological Spaces"

Let X be a topological_space.

Definition 1.

Let Y be a topological_space. Let M be a subset of Y. M is closed IFF for any point_ Q of Y it_is_true_that if P is a point of $ST(M)$ infinitesimally close to Q then Q is a point of M.

Definition 2.

Let Y be a topological_space. Let M be a subset of Y. M is compact IFF for any point_ P of $ST(M)$ there exists Q such that Q is a point of M and P is infinitesimally close to Q.

Definition 3.

Let Y be a set. P is a point of Y IFF P is an element of Y.

Lemma 6.

Any topological_space is a set.

Theorem.

Any closed subset of a compact set is compact.

After the syntactical transformation of the above TL-text, we get the following $TL1$-text.

1. "Sets and Relations"

Lemma 1. Let V be a set. Let W be a subset of V. W is a set.

Lemma 2. Let V be a set. Let U be a subset of V. If t is an element of U then t is an element of V.

Lemma 3. Let Y be a set. $ST(Y)$ is a set.

Lemma 4. Let U be a subset of X. Let V be a subset of X. If U is a subset of V then $ST(U)$ is a subset of $ST(V)$.

Lemma 5. "The transitivity property of inclusion"

For any U, V, W it_is_true_that if U is a subset of V and V is a subset of W then U is a subset of W.

2. "Topological Spaces"

Let X be a topological_space.

Definition 1.

Let Y be a topological_space. Let M be a subset of Y.
If M is closed then for any Q it_is_true_that (if Q is a point of Y then for any P it_is_true_that (if P is a point of $ST(M)$ and P is infinitesimally close to Q then Q is a point of M)).

Let Y be a topological_space. Let M be a subset of Y.
If for any Q it_is_true_that (if Q is a point of Y then for any P it_is_true_that (if P is a point of $ST(M)$ and P is infinitesimally close to Q then Q is a point of M)) then M is closed.

Definition 2.

Let Y be a topological_space. Let M be a subset of Y.
If M is compact then for any P it_is_true_that (if P is a point of $ST(M)$ then there exists Q such that (Q is a point of M and P is infinitesimally close to Q)).

Let Y be a topological_space. Let M be a subset of Y.
If for any P it_is_true_that (if P is a point of $ST(M)$ then there exists Q such that (Q is a point of M and P is infinitesimally close to Q)) then M is compact.

Definition 3.

Let Y be a set.
If P is a point of Y then P is an element of Y.

Let Y is a set.
If P is an element of Y then P is a point of Y.

Lemma 6.

Let X be a topological_space. X is a set.

Theorem.

Let X be a topological space. Let K be a subset of X. Let F be a subset of K. Let K be compact. Let F be closed. Then F is compact.

Proof.

The following initial sequent corresponds to the theorem.

$[\mathcal{D}], [\mathcal{P}], [\,], <, \overline{X}$ is a_T_S $\wedge \overline{K}$ is a_subset of $\overline{X} \wedge \overline{F}$ is a_subset of $\overline{K} \wedge \overline{K}$ is compact $\wedge \overline{F}$ is closed, $> \Rightarrow <, \overline{F}$ is compact, $>$

Here we use the abbreviations T_S and $\approx$ for "topological_space" and "infinitesimally close", respectively. The lists $[\mathcal{D}]$ and $[\mathcal{P}]$ constitute proof environment and consist of all the definitions and lemmas, respectively, from the above $TL1$-text. In what follows, we omit $[\mathcal{D}]$ and $[\mathcal{P}]$ but point out those definitions and auxiliary propositions to which DA and AP rules are applied. Note, that when DA is applied, a definition is copied and then those variables which belong to the copy are renamed. In the text below, each renamed variable has a superscript which is equal to the number of a particular definition.

Let Π denote the premises "$\overline{X}$ is a T_S, $\overline{K}$ is a subset of $\overline{X}$, $\overline{F}$ is a subset of $\overline{K}$, $\overline{K}$ is compact, $\overline{F}$ is closed" in the antecedent of the sequent above (overlined variables are fixed variables). The DA rule is applicable to Definition 2. Two sequents are generated:

1. $[\sim \overline{F}$ is compact$], \Pi \Rightarrow Z_1^2$ is a T_S $\wedge Z_2^2$ is a subset of $\overline{X}$

2. $\Pi, \forall Z_3^2(Z_3^2$ is a point of $ST(Z_2^2) \supset \exists Z_4^2(Z_4^2$ is a point of $Z_2^2 \wedge Z_3^2 \approx Z_4^2)) \supset Z_2^2$ is compact $\Rightarrow \overline{F}$ is compact

We exploit the dual nature of $TL1$-sentences and use standard logical connectives to represent $TL1$-sentences in a shorthand form. Note, that applying quantifier rules, we do not show duplicated formulas to save room. Applying GS rule to 1, we have new sequents

1.1. $[\sim \overline{F}$ is compact$], \Pi \Rightarrow Z_1^2$ is a T_S

1.2. $[\sim \overline{F}$ is compact$], \Pi \Rightarrow Z_2^2$ is a subset of $\overline{X}$

The sequent 1.1 is provable, and we have a substitution $\{\overline{X}/Z_1^2\}$. Applying AG rule to 2, we have

2.1. Π, Z_2^2 is compact $\Rightarrow \overline{F}$ is compact

2.2. $[\sim \overline{F}$ is compact$], \Pi \Rightarrow \forall Z_3^2(Z_3^2$ is a point of $ST(Z_2^2) \supset \exists Z_4^2(Z_4^2$ is a point of $Z_2^2 \wedge Z_3^2 \approx Z_4^2))$

The sequent 2.1. is provable, and the corresponding substitution is $\{\overline{F}/Z_2^2\}$. Then applying GS rules to 2.2, we obtain

2.2.1. $[\sim \overline{F}$ is compact$], \Pi, \overline{Z}_3^2$ is a point of $ST(\overline{F}), \forall Z_4^2 \neg(Z_4^2$ is a point of $\overline{F} \wedge \overline{Z}_3^2 \approx Z_4^2) \Rightarrow Z_4^2$ is a point of $\overline{F}$

2.2.2. $[\sim \overline{F}$ is compact$], \Pi, \overline{Z}_3^2$ is a point of $ST(\overline{F}), \forall Z_4^2 \neg(Z_4^2$ is a point of $\overline{F} \wedge \overline{Z}_3^2 \approx Z_4^2) \Rightarrow \overline{Z}_3^2 \approx Z_4^2$

Let us consider the sequent 2.2.2. Denote the collection of the premises of 2.2.2 by Π_1. The rule DA is applicable to Definition 2. Processing 2.2.2

(by DA and then by AG), we have the following sequents:

2.2.2.1. $[\sim \overline{F}$ is compact, $\sim \overline{Z}_3^2 \approx Z_4^2], \Pi_1 \Rightarrow Z_5^2$ is a T_S

2.2.2.2. $[\sim \overline{F}$ is compact, $\sim \overline{Z}_3^2 \approx Z_4^2], \Pi_1 \Rightarrow Z_6^2$ is a subset of $\overline{X}$

2.2.2.3. $[\sim \overline{F}$ is compact, $\sim \overline{Z}_3^2 \approx Z_4^2], \Pi_1 \Rightarrow Z_7^2$ is a point of $ST(\overline{K})$

2.2.2.4. $[\sim \overline{F}$ is compact$], \Pi_1, \overline{Z}_8^2$ is a point of $\overline{K}, Z_7^2 \approx \overline{Z}_8^2 \Rightarrow \overline{Z}_3^2 \approx Z_4^2$

The sequents 2.2.2.1, 2.2.2.2, 2.2.2.4 are provable, the corresponding (common) substitution is $\{\overline{X}/Z_5^2, \overline{K}/Z_6^2, \overline{Z}_3^2/Z_7^2, \overline{Z}_8^2/Z_4^2\}$. Note, that there are two new premises now: $\overline{Z}_8^2$ is a point of $\overline{K}$ and $\overline{Z}_3^2 \approx \overline{Z}_8^2$. Taking into account above new premises (i.e. applying PA rule) and the substitution generated, we can write down the sequent 2.2.1 in the following form:
$[\sim \overline{F}$ is compact$], \Pi_1, \overline{Z}_8^2$ is a point of $\overline{K}, \overline{Z}_3^2 \approx \overline{Z}_8^2 \Rightarrow \overline{Z}_8^2$ is a point of $\overline{F}$.

Now DA is applicable to Definition 1. The chain of DA, AG, and GS applications gives:

2.2.1.1. $[\sim \overline{F}$ is compact, $\sim \overline{Z}_8^2$ is a point of $\overline{F}], \Pi_1, \overline{Z}_8^2$ is a point of $\overline{K}, \overline{Z}_3^2 \approx \overline{Z}_8^2 \Rightarrow Z_1^1$ is a T_S

2.2.1.2. $[\sim \overline{F}$ is compact, $\sim \overline{Z}_8^2$ is a point of $\overline{F}], \Pi_1, \overline{Z}_8^2$ is a point of $\overline{K}, \overline{Z}_3^2 \approx \overline{Z}_8^2 \Rightarrow Z_2^1$ is a subset of $\overline{X}$

2.2.1.3. $[\sim \overline{F}$ is compact, $\sim \overline{Z}_8^2$ is a point of $\overline{F}], \Pi_1, \overline{Z}_8^2$ is a point of $\overline{K}, \overline{Z}_3^2 \approx \overline{Z}_8^2 \Rightarrow Z_3^1$ is a point of $\overline{X}$

2.2.1.4. $[\sim \overline{F}$ is compact$], \Pi_1, \overline{Z}_8^2$ is a point of $\overline{K}, \overline{Z}_3^2 \approx \overline{Z}_8^2, Z_3^1$ is a point of $Z_2^1 \Rightarrow \overline{Z}_8^2$ is a point of $\overline{F}$

2.2.1.5. $[\sim \overline{F}$ is compact, $\sim \overline{Z}_8^2$ is a point of $\overline{F}], \Pi_1, \overline{Z}_8^2$ is a point of $\overline{K}, \overline{Z}_3^2 \approx \overline{Z}_8^2 \Rightarrow Z_4^1 \approx \overline{Z}_8^2$

2.2.1.6. $[\sim \overline{F}$ is compact, $\sim \overline{Z}_8^2$ is a point of $\overline{F}], \Pi_1, \overline{Z}_8^2$ is a point of $\overline{K}, \overline{Z}_3^2 \approx \overline{Z}_8^2 \Rightarrow Z_4^1$ is a point of $ST(\overline{F})$

The sequents 2.2.1.1, 2.2.1.4, 2.2.1.5, 2.2.1.6 are provable, the substitution generated is $\{\overline{X}/Z_1^1, \overline{Z}_8^2/Z_3^1, \overline{F}/Z_2^1, \overline{Z}_3^2/Z_4^1\}$.

Let us consider the sequent 1.2. It can be written now as follows:

$[\sim \overline{F}$ is compact$], \Pi \Rightarrow \overline{F}$ is a subset of $\overline{X}$. Applying AP to Lemma 5 (the transitivity of inclusion) and then AG rules, we have:

1.2.1. $[\sim \overline{F}$ is compact, $\sim \overline{F}$ is a subset of $\overline{X}], \Pi \Rightarrow Z_1^t$ is a subset of Z_2^t

1.2.2. $[\sim \overline{F}$ is compact, $\sim \overline{F}$ is a subset of $\overline{X}], \Pi \Rightarrow Z_2^t$ is a subset of Z_3^t

1.2.3. $[\sim \overline{F}$ is compact$], \Pi, Z_1^t$ is a subset of $Z_3^t \Rightarrow \overline{F}$ is a subset of $\overline{X}$.

The sequent 1.2 is provable ($\{\overline{F}/Z_1^t, \overline{K}/Z_2^t, \overline{X}/Z_3^t\}$. The sequent 2.2.1.2 can be proved similarly.

We proceed processing 2.2.2.3. Applying the substitution generated we have $[\sim \overline{F}$ is compact,$\sim \overline{Z}_3^2 \approx \overline{Z}_8^2], \Pi_1 \Rightarrow \overline{Z}_3^2$ is a point of $ST(\overline{K})$. Notice, that Π_1 contains the premise $\overline{Z}_3^2$ is a point of $ST(\overline{F})$. Denote the antecedent of the sequent by $[B_1], \Pi_1$. The rule DA is applicable to Definition 3. We obtain:

2.2.2.3.1. $[B_1, \sim \overline{Z}_3^2$ is a point of $ST(\overline{K})], \Pi_1 \Rightarrow Z_1^3$ is a set

2.2.2.3.2. $[B_1, \sim \overline{Z}_3^2$ is a point of $ST(\overline{K})], \Pi_1 \Rightarrow Z_2^3$ is an element of Z_1^3

2.2.2.3.3. $[B_1], \Pi_1, Z_2^3$ is a point of $Z_1^3 \Rightarrow \overline{Z}_3^2$ is a point of $ST(\overline{K})$

The sequent 2.2.2.3.3 is provable, the substitution generated is $\{\overline{Z}_3^2/Z_2^3, ST(\overline{K})/Z_1^3\}$. Then 2.2.2.3.1 is of the form:

$[B_1, \sim \overline{Z}_3^2$ is a point of $ST(\overline{K})], \Pi_1 \Rightarrow ST(\overline{K})$ is a set

The sequent 2.2.2.3.2 is now of the form: $[B_1, \sim \overline{Z}_3^2$ is a point of $ST(\overline{K})]$, $\Pi_1 \Rightarrow \overline{Z}_3^2$ is an element of $ST(\overline{K})$.

Let us consider the first of these two sequents. Denote its antecedent by $[B_2], \Pi_2$. The rule AP is applicable to Lemma 3.

2.2.2.3.1.1. $[B_2, \sim ST(\overline{K})$ is a set$], \Pi_2 \Rightarrow Z_1^{l3}$ is a set

2.2.2.3.1.2. $[B_2], \Pi_2, ST(Z_1^{l3})$ is a set $\Rightarrow ST(\overline{K})$ is a set

The sequent 2.2.2.3.1.2 is provable ($\{\overline{K}/Z_1^{l3}\}$). The rule AP is applicable to 2.2.2.3.1.1 and Lemma 1 giving

2.2.2.3.1.1.1. $[B_2, \sim ST(\overline{K})$ is a set,$\sim \overline{K}$ is a set$], \Pi_2 \Rightarrow Z_1^{l1}$ is a set

2.2.2.3.1.1.2. $[B_2, \sim ST(\overline{K})$ is a set, $\sim \overline{K}$ is a set $], \Pi_2 \Rightarrow Z_2^{l3}$ is a subset of Z_1^{l1}

2.2.2.3.1.1.3. $[B_2, \sim ST(\overline{K})$ is a set$], \Pi_2, Z_2^{l1}$ is a set $\Rightarrow \overline{K}$ is a set

The sequent 2.2.2.3.1.1.2 is provable ($\{\overline{K}/Z_2^{l1}, \overline{X}/Z_1^{l1}\}$).

Substituting in 2.2.2.3.1.1.1 we have:

$[B_2, \sim ST(\overline{K})$ is a set,$\sim \overline{K}$ is a set$], \Pi_2 \Rightarrow \overline{X}$ is a set

Applying AP to Lemma 6 gives:

2.2.2.3.1.1.1.1. $[B_2, \sim ST(\overline{K})$ is a set,$\sim \overline{K}$ is a set,$\sim \overline{X}$ is a set$], \Pi_2 \Rightarrow Z_1^{l6}$ is a T_S

2.2.2.3.1.1.1.2. $[B_2, \sim ST(\overline{K})$ is a set,$\sim \overline{K}$ is a set$], \Pi_2, Z_1^{l6}$ is a set $\Rightarrow \overline{X}$ is a set

The above sequents are provable ($\{\overline{X}/Z_1^{l6}\}$). Denote the antecedent of 2.2.2.3.2 by $[B_3], \Pi_3$. Applying AP to Lemma 3 we obtain

2.2.2.3.2.1. $[B_3, \sim \overline{Z}_3^2$ is an element of $ST(\overline{K})], \Pi_3 \Rightarrow Z_1^{l2}$ is a subset of Z_2^{l2}

2.2.2.3.2.2. $[B_3, \sim \overline{Z}_3^2$ is an element of $ST(\overline{K})], \Pi_3 \Rightarrow Z_3^{l2}$ is an element of Z_1^{l2}

2.2.2.3.2.3. $[B_3], \Pi_3, Z_3^{l2}$ is an element of $Z_2^{l2} \Rightarrow \overline{Z}_3^2$ is an element of $ST(\overline{K})$

The sequent 2.2.2.3.2.3 is provable ($\{ST(\overline{K})/Z_2^{l2}, \overline{Z}_3^2/Z_3^{l2}\}$). The application of AP to 2.2.2.3.2.1 and Lemma 2 produces

2.2.2.3.2.1.1. $[B_3, \sim \overline{Z}_3^2$ is an element of $ST(\overline{K}), Z_1^{l2}$ is a subset of $ST(\overline{K})], \Pi_3 \Rightarrow Z_1^{l4}$ is a subset of Z_2^{l4}

2.2.2.3.2.1.2. $[B_3, \sim \overline{Z}_3^2$ is an element of $ST(\overline{K})], \Pi_3, ST(Z_1^{l4})$ is a subset of $ST(Z_2^{l4}) \Rightarrow Z_1^{l2}$ is a subset of $ST(\overline{K})$

which are provable $(\{\overline{F}/Z_1^{l4}, \overline{K}/Z_2^{l4}, ST(\overline{F})/Z_1^{l2}\})$.

To prove 2.2.2.3.2.2, we apply DA to Definition 3.

2.2.2.3.2.2.1. $[B_3, \sim \overline{Z}_3^2$ is an element of $ST(\overline{K}), \sim \overline{Z}_3^2$ is an element of $ST(\overline{F})], \Pi_3 \Rightarrow Z_3^3$ is a set

2.2.2.3.2.2.2. $[B_3, \sim \overline{Z}_3^2$ is an element of $ST(\overline{K}), \sim \overline{Z}_3^2$ is an element of $ST(\overline{F})], \Pi_3 \Rightarrow Z_4^3$ is a point of Z_3^3

2.2.2.3.2.2.3. $[B_3, \sim \overline{Z}_3^2$ is an element of $ST(\overline{K})], \Pi_3, Z_4^3$ is an element of $Z_3^3 \Rightarrow \overline{Z}_3^2$ is an element of $ST(\overline{F})$

The seguents 2.2.2.3.2.2.2 and 2.2.2.3.2.2.3 are provable $(\{ST(\overline{F})/Z_3^3, \overline{Z}_3^2/Z_4^3\})$. The application of AP to 2.2.2.3.2.2.1 and Lemma 3 gives

2.2.2.3.2.2.1.1. $[B_3, \sim \overline{Z}_3^2$ is an element of $ST(\overline{K}), \sim \overline{Z}_3^2$ is an element of $ST(\overline{F}), \sim ST(\overline{F})$ is a set$], \Pi_3 \Rightarrow Z_2^{l3}$ is a set

2.2.2.3.2.2.1.2. $[B_3, \sim \overline{Z}_3^2$ is an element of $ST(\overline{K}), \sim \overline{Z}_3^2$ is an element of $ST(\overline{F})], \Pi_3, ST(Z_2^{l3})$ is a set $\Rightarrow ST(\overline{F})$ is a set

The sequent 2.2.2.3.2.2.1.2 is proved $(\{\overline{F}/Z_2^{l3}\})$.

To prove 2.2.2.3.2.2.1.1 the rule AP should be applied to Lemma 1. The sequent 2.2.1.3 is provable by the application of DA (to Definition 3), AP (to Lemma 2), and then DA (to Definition 3).

Of course, the above mS inference of the theorem requires editing to take the form of a proof from a usual mathematical publication. Constructing the edition of the proof, it should be reasonable to "hide" the inference steps concerning, for example, going from the notion of a point to the notion of an element and set-theoretical reasoning steps.